

Home Screen

The screenshot displays a web application interface. At the top, a browser address bar shows the URL `https://reactjs-poc-ui-master.herokuapp.com/dashboard`. The application's header includes a navigation menu on the left with items: Dashboard, Create Risk, View Risk, and About Me. The main content area features a greeting: "Hi, admin" and "Roles: admin", followed by a note: "This site is best viewed with latest Firefox or Chrome with 80% zoom." Below this, a "Create Risk Type" form is shown, which is a nested view of the application. This nested view includes a sidebar with "Dashboard" (marked as "NEW"), "Create Risk", "Risk Type", "Risk Instance", and "View Risk". The main form area has input fields for "Name" (containing "Automobile") and "Description" (containing "Risk type for Automobile"), an "Add Type Field" button, and a table with the following data:

Field Name	Field Description	Field Enum	Action
model	Model of Automobile	text	<button>Edit</button> <button>Delete</button>

ReactJS PoC UI

- <https://reactjs-poc-ui-master.herokuapp.com/login>
- Use above link to access UI.
- Username :- admin (lower case)
- Password :- poctest#1 (lower case)
- Best viewed with Firefox with 80 % zoom
- Above UI refers Web API at following location credentials are same.
- <https://django-poc-server-maheshbodas.herokuapp.com/>

Salient features

- Developed using ReactJS, Redux, redux-thunk, ES6, Element React library, Node JS, Serve as a Static server.
- Used Redux as global data store for this application. Redux actions and reducers make it easy to shape and store data as needed by UI.
- Used CoreUI React template so as to reuse Sidebar navigation, Breadcrumbs and common Layout for all pages.

Salient features (continued)

- CoreUI React template itself build using Create React App. CRA is zero configuration template to build, unit test and deploy application on server.
- CRA makes use of Enzyme and Jest for Unit testing. Does not need separate Jest installation or Web pack tweaking. Everything is taken care by CRA.

Salient features (continued)

- Well thought React components to modularize code.
- Made use of Typescript and React Typescript support to develop generic component that's used in all screen to automate task like showing Page loading Icon, display UI when data is available and show error if any.
- The above component has been reused in all screens of ReactJS PoC application.

Salient features (continued)

- Role based authorization further limits access to screens and components use for creating RiskTypes.
- User can add RiskType and associated RiskTypeFields in one go using header and details table in screen.
- User can add Risk and Risk Fields in one go using header and details table in screen.

Salient features (continued)


- Create Risk Instance screen collect data by dynamically adding controls related to various Risk type fields.
- Create Risk Instance screen dynamically adds required field validation and field type specific validation for controls related to Risk type fields. i.e. Date / float / integer.
- Successful model creation messages are displayed using MessageBox.
- Display various model validation errors using MessageBox



Salient features (continued)

- View Single Risk allows user to fetch details for given Risk and renders various read-only controls related to Risk Type fields.
- View All Risks allow user to view All Risks entered in system based on RiskType filter.

Login Screen

VueJS Poc UI User

 mahesh.bodas

 •••••••• 

Sign in

username: admin password: poctest#1

Dashboard showing app highlights

The dashboard features a top navigation bar with a logo, a hamburger menu, the title 'Dashboard', a user profile icon, and another hamburger menu. A left sidebar contains links to 'Dashboard', 'Create Risk', 'View Risk', and 'About Me'. The main content area displays a welcome message: 'Hi, admin' and 'Roles: admin', followed by a note: 'This site is best viewed with latest Firefox or Chrome with 80% zoom.' Below this is a preview of the 'Create Risk Type' form, which includes input fields for 'Name' (containing 'Automobile') and 'Description' (containing 'Risk type for Automobile'), an 'Add Type Field' button, a table of existing fields, and a 'Submit' button. The table has columns for 'Field Name', 'Field Description', 'Field Enum', and 'Action'. The right sidebar lists application features: 'ReactJS PoC', 'Powered by ReactJS ...', 'Development', 'Technology used', 'Unit testing', 'Tools used', and 'Authentication', each with a brief description and a user profile icon.

Dashboard

Hi, admin
Roles: admin

This site is best viewed with latest Firefox or Chrome with 80% zoom.

Create Risk Type

Dashboard / CreateRiskTypeCtrl / CreateRiskTypeCtrl

* Name: Automobile Description: Risk type for Automobile

Add Type Field

Field Name	Field Description	Field Enum	Action
model	Model of Automobile	test	Edit Delete
doors	Number of door	integer	Edit Delete

Submit

ReactJS PoC

Powered by ReactJS ...
An application to Create and View user defined entities. Implement dynamic forms with validations for entity creation.

Development

Technology used
Developed using ReactJS, ES6, Element UI library, Node JS, Express server. Redux and Redux-Thunk as a middleware.

Unit testing

Tools used
Used Jest, Enzyme and Redux Mock Store for Unit testing of React Components, Redux actions, Redux reducers.

Authentication


Role based authoriza...
Allows only admin user to access all pages in application. Role based authorization

Create Risk Type screen

- Admin User can define Risk types.
- User Enters Risk type name and description.
- Click Add Risk Type Field button to define Risk type fields.
- Post Risk type after adding / editing Risk type fields.

Create Risk Type Image

← → ↻ <https://reactjs-poc-ui-master.herokuapp.com/create/creatertypectrl> 🔑 🔍 ☆ 🚫 📄 ⚙️ | Paused 👤 ⋮

⚙️ ☁️ Dashboard  ⋮

[Dashboard](#)[Create Risk](#)[Risk Type](#)[Risk Instance](#)[View Risk](#)[About Me](#)

CreateRiskTypeCtrl / CreateRiskTypeCtrl

Field Name	Field Description	Field Enum	Action
houzenumber	House number allotted by corporation	text	Edit Delete
floors	Number of floors	integer	Edit Delete
sum	Sum Insurance Amount	currency	Edit Delete
completion	Construction completion date	date	Edit Delete

[Submit](#)

[Hire me](#) © 2019 Mahesh Bodas. Powered by [ReactJS](#)

Risk Type Data posted to server

```
{"risk_type_name":"Home","risk_type_description":"Risk type for Home","risktype_risktypefields":[{"risk_type_field_name":"houzenumber","risk_type_field_enum":"text","risk_type_field_description":"House Number"}, {"risk_type_field_name":"floors","risk_type_field_enum":"integer","risk_type_field_description":"Number of floors"}, {"risk_type_field_name":"sum","risk_type_field_enum":"currency","risk_type_field_description":"Insurance amount"}, {"risk_type_field_name":"completion","risk_type_field_enum":"date","risk_type_field_description":"Date of completion"}]}
```

Create RiskType shows Success

The screenshot shows a web browser at the URL `https://reactjs-poc-ui-master.herokuapp.com/create/createrisktypectrl`. The application has a dark sidebar with navigation links: Dashboard, Create Risk, Risk Type, Risk Instance, View Risk, and About Me. The main content area is titled 'CreateRiskTypeCtrl / CreateRiskTypeCtrl' and contains a form with two input fields: 'Name' (with a red asterisk) and 'Description'. Below these fields are two blue buttons: 'Add Type Field' and 'Submit'. A success notification box in the top right corner displays a green checkmark and the text 'Success Risk Type created successfully'. The footer includes the text 'Hire me © 2019 Mahesh Bodas.' and 'Powered by ReactJS'.

Dashboard

CreateRiskTypeCtrl / CreateRiskTypeCtrl

* Name Description

Add Type Field

Submit

Success
Risk Type created successfully

Hire me © 2019 Mahesh Bodas. Powered by ReactJS

Create Risk Instance screen.

- User can create Risk Instance based on Risk types.
- Select appropriate Risk Type from dropdown box.
- Fill up all Risk Instance fields and submit form.
- Admin user can define Risk Types.

Screen showing field validation errors

Browser address bar: <https://reactjs-poc-ui-master.herokuapp.com/create/createtrickctrl>

Dashboard

CreateRiskTypeCtrl / CreateRiskCtrl

Select Risk Type: Automobile

<p>* Risk Name</p> <input type="text" value="risk_name"/> <p>risk_name is required</p>	<p>Risk Description</p> <input type="text" value="risk_description"/>
<p>Amount</p> <input type="text" value="\$0.00"/>	<p>* Doors</p> <input type="text" value="0"/> <p>doors is required</p>
<p>* Issuedate</p> <input type="text" value="issuedate"/> <p>Issuedate is required</p>	<p>* Model</p> <input type="text" value="model"/> <p>model is required</p>

Submit when validation succeed

Browser address bar: <https://reactjs-poc-ui-master.herokuapp.com/create/createriskctrl>

Page title: Dashboard

Navigation menu:

- Dashboard
- Create Risk
- Risk Type
- Risk Instance
- View Risk
- About Me

Form content:

Select Risk Type: Automobile

* Risk Name:

Risk Description:

Amount:

* Doors:

* Issuedate:

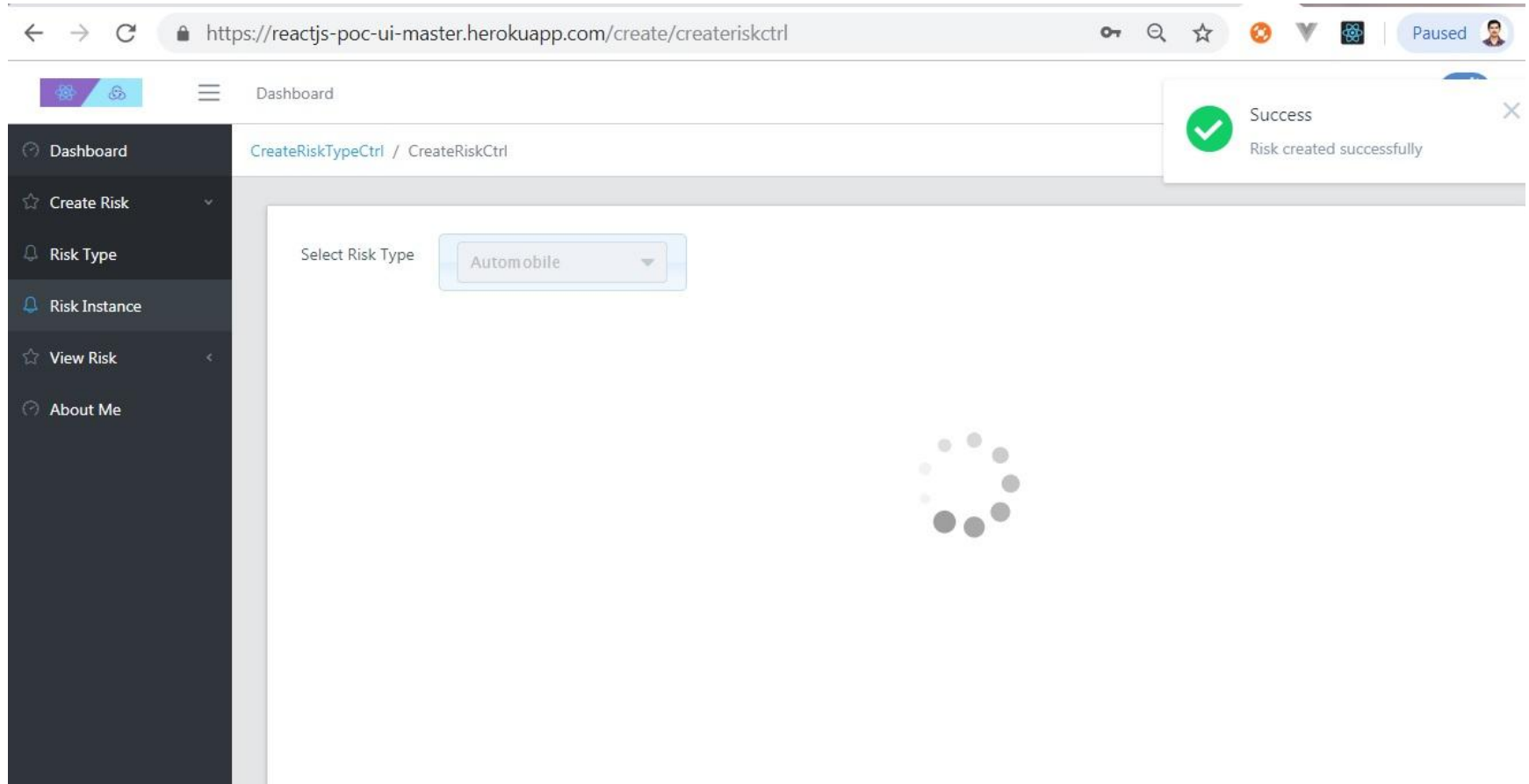
* Model:

Buttons:

Posting Risk Instance data to server

```
{"risktype":1,"risk_name":"Mercedes 1","risk_description":"Insurance for Mercedes","risk_riskfields":[{"risktypefield":4,"risk_field_value":"07/14/2016"},{"risktypefield":3,"risk_field_value":30000},{"risktypefield":2,"risk_field_value":4},{"risktypefield":1,"risk_field_value":"Mercedes"}]}
```

Create Risk Instance showing success



View Single Risk Instance

- User can view single Risk Instance using this form.
- Select appropriate Risk Instance from dropdown box.
- Screen will show Risk name and all Risk Instance fields (read-only).

View Single Risk Screen

← → ↻ <https://reactjs-poc-ui-master.herokuapp.com/view/viewriskctrl> 🔑 🔍 ☆ 🚫 📄 🧠 | Paused 👤 ⋮

⚙️ 🌐 Dashboard

ViewRiskCtrl / ViewRiskCtrl

🕒 Dashboard

☆ Create Risk <

☆ View Risk ▾

🔔 Single Risk

🔔 All Risks

🕒 About Me

Select Risk

Risk Name	Risk Description
<input type="text" value="Mercedes 1"/>	<input type="text" value="Risk policy for Mercedes"/>
Amount	Doors
<input type="text" value="\$30,000.00"/>	<input type="text" value="4"/> - +
Issuedate	Model
<input type="text" value="09/14/2017"/> 📅	<input type="text" value="Mercedes"/>

View All Risks

- User can view Risk Instances using this form
- Select appropriate Risk Type from dropdown box.
- Screen will show all Risks and associated instance fields in tabular format.

View All Risk Screen

Browser address bar: <https://reactjs-poc-ui-master.herokuapp.com/view/viewriskgrid>

Page Header: Dashboard

Left Sidebar:

- Dashboard
- Create Risk
- View Risk
- Single Risk
- All Risks
- About Me

Breadcrumb: ViewRiskCtrl / ViewRiskGrid

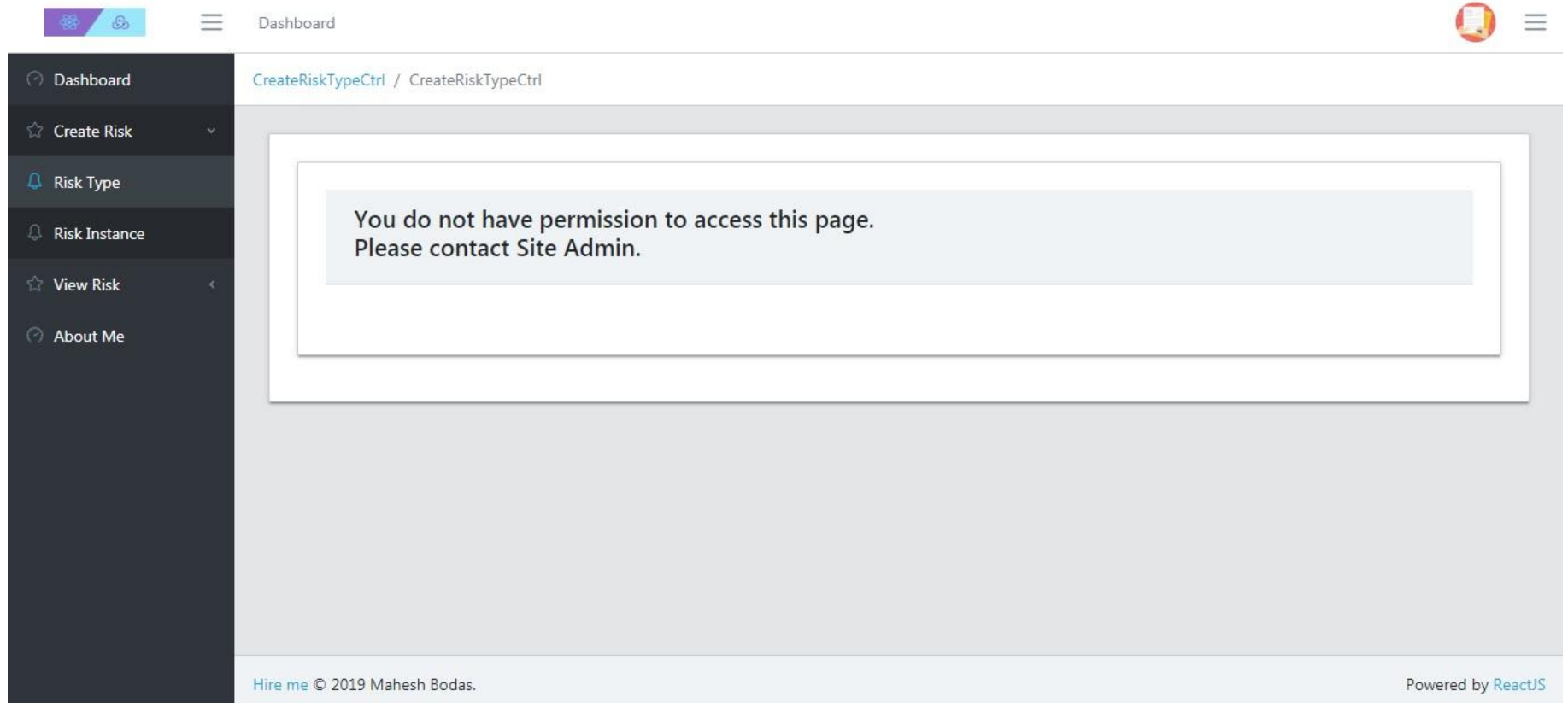
Select Risk Type:

Risk Name	Risk Description	Amount	Doors	Issuedate	Model
Mercedes 1	Risk policy for Mercedes	\$ 30,000.00	4	09/14/2017	Mercedes
Toyota 1	Risk policy for Toyota 1	\$ 10,000.00	4	07/22/2016	Toyota

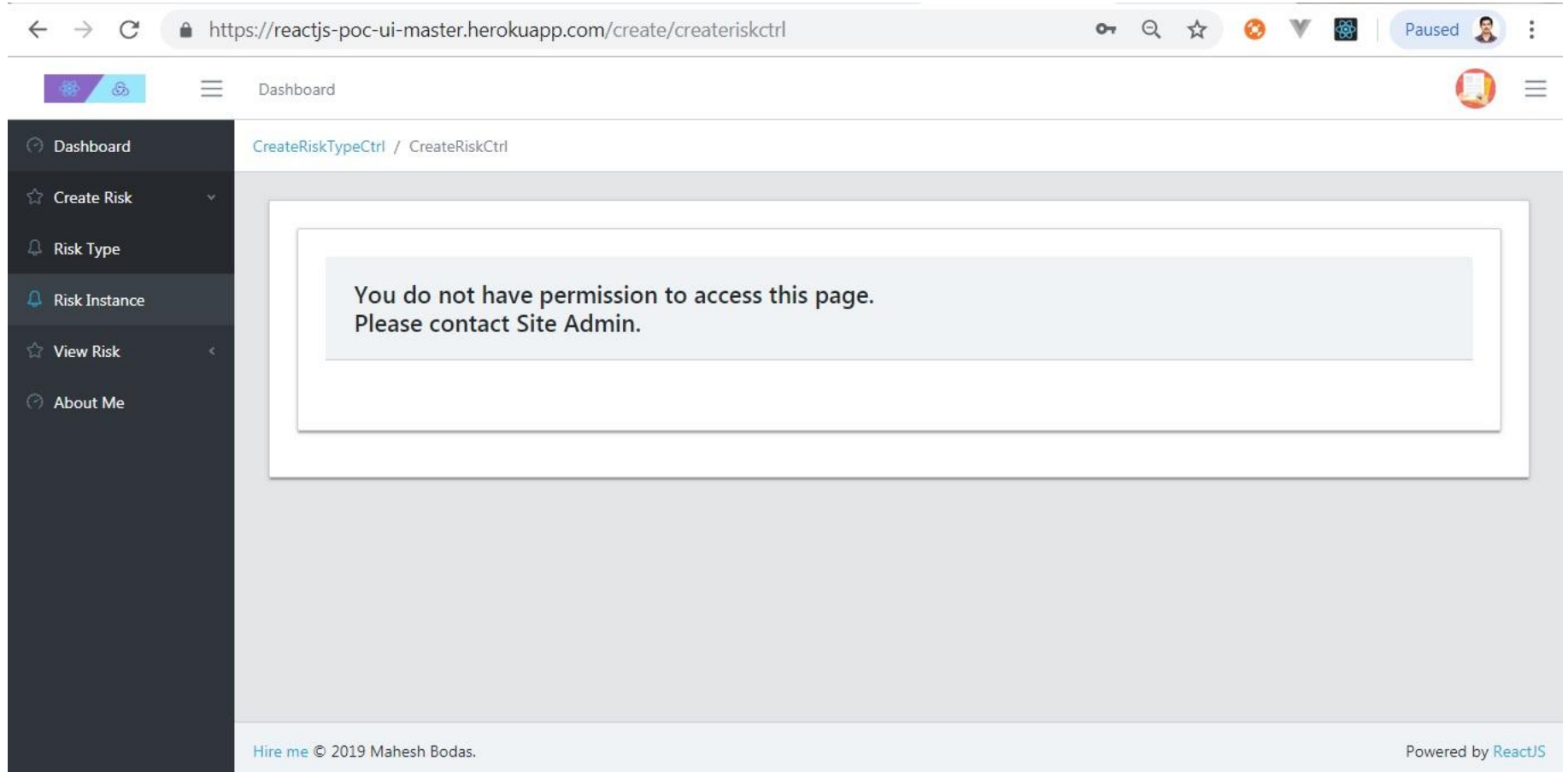
Non Admin users

- All above slides assumed user logged in has admin privilege.
- Non admin user will be restricted from accessing Create Risk Type and Create Risk Instance page.
- They can view Risk Instances added to system.
- User : editor (lower case)
- Password : test#123 (lower case)

No access to Create Risk Type



No access to Create Risk Instance



Unit testing.

- Used Enzyme, Jest for unit testing React components.
- Used Redux Mock Store for Unit testing of, Redux actions, Redux reducers.

Known issues.

- Since Django Rest API and UI application are installed using free account on Heroku platform.
- In case if Django Rest API is not invoked in last one hour or more Dyno associated with it goes to sleep.
- As a result it may be possible that when UI invokes API it might timeout.
- In case if you get any error try refreshing page after 2/3 minutes.

Known issue (continued)

